UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/696,467	10/28/2003	Yu Kwong Ng	2705-726	6905

20575 7590 11/06/2007

MARGER JOHNSON & MCCOLLOM, P.C.
210 SW MORRISON STREET, SUITE 400
PORTLAND, OR 97204

EXAMINER	
WALTER, CRAIG E	

ART UNIT	PAPER NUMBER
2188	

MAIL DATE	DELIVERY MODE
11/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/696,467	NG ET AL.	
	Examiner	Art Unit	
	Craig E. Walter	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 September 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14, 27-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14, 27-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Status of Claims

1. Claims 1-14 and 27-38 are pending in the Application.

Claims 15-26 remain cancelled.

Claims 1, 2, 6-9, 11-14, 27, 28, 30, 31, and 34-38 are amended.

Claims 1-14 and 27-38 are rejected.

Response to Amendment

2. Applicant's amendments and arguments filed on 19 September 2007 in response to the office action mailed on 19 June 2007 have been fully considered, but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claim 36 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the phrase "partial key includes multiple consecutive sequential strings of bits derived from the input key, separated by the bits of the hash value" renders the claim indefinite. More specifically, does this recitation refer to the partial key itself (which contains the multiple bits) as being separated from the input key, or the bits themselves contained within the partial key as

being separated by the bits of the hash value? Which is being claimed here? The former will be assumed for the purposes of applying art.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-4, 7-10, 27-30 and 34-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin et al. (US Patent 6,493,813 B1), hereinafter Brandin (US Patent 6,493,813) in further view of Biran (US Patent 6,345,347).

As for claims 1 and 27, Brandin teaches:

storing a plurality of partial keys (Fig 13b illustrates a partial key (element 324, A+B) corresponding to an equal number of original keys in a hash table (each original key A+B+C, contains one and only one partial key (A+B)), wherein storage of the plurality of partial keys requires less memory than storage of the equal number of original keys (128 bit partial key compared to a 196 bit original key), and wherein the plurality of partial keys are used to determine hashing conflicts (element 324). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). Each key is split into a partial key as illustrated in Fig. 13b), and are subsequently provided to the transform generator – col. 2, lines 54-57.

Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys)). Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a partial key (i.e. Fig. 13B). Each partial key (i.e. 128-bit partial key) form the constituent elements of, and correspond to, its respective original key (196-bits). The partial keys (158-bit) require less memory than the original keys (196-bits);

applying a hash function to an original key of said equal number of original keys to generate a partial key and a hash value, wherein the hash value includes a number of bits equal to a number of bits of the original key minus a number of bits of the partial key (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to the partial key in Fig 13b.). Additionally note, the claim requires the hash value to equal "a number" of bits of the original key minus "a number" of bits of the partial key. Note "a number" is broadly construed as "any number" rather than "total number" of bits. Based on Examiner's broadest reasonable interpretation pursuant to MPEP § 2111, the number of Brandin's hash value equals a number of bits in the original key minus a number of bits in the partial key (e.g. all 196 bits of the original key minus 0 bits of the partial key = 196 which is the size of the hash value);

accessing the hash table according to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49);

reading a stored partial key of the plurality of partial keys from the hash table that corresponds to the hash value, wherein said hash value is based on said original key (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11). The entry containing the partial key is read in order to obtain this information. Again, Fig. 13b illustrates that this can be applied to a partial key if the original key is greater than 64 bits;

Note giving the limitation "reading a partial key from the memory that corresponds to said hash value, wherein said hash value is based on said original key" its "broadest reasonable interpretation consistent with the specification" (In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)), reading a partial key may include reading the entire key (which is split into a partial key), since the each partial key is inherently read during the step of reading the key in its entirety. The hash value will always be "based on the original key" even if said value is only made up by a portion of the original key (i.e. the portion corresponding to the partial key).

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a stored partial key stored in the memory, wherein the partial key corresponds at most with one of the stored partial keys ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). Note each 196-bit key contains 128 bits (one-one

correspondence). He fails to teach however, actually comparing the keys (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys). Additionally, Brandin fails to teach not storing the hash value in the hash table (Brandin rather teaches storing the value in the hash table).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)). Additionally, Biran teaches managing the pointer and hash value functionality in the I/O adapter (Fig. 3, element 38 – translation table), rather than in the system memory (i.e. not in the memory).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

As for claims 7 and 34, Brandin teaches an apparatus and system, comprising:

a hash table which stores a plurality of partial keys used to determine hashing conflicts, wherein the plurality of partial keys correspond to a plurality of

original full keys (Fig. 13b illustrates a partial key (the 128-bit portion of depicted as elements A and B in element 324). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). Each key is split into a partial key (elements A+B) as illustrated in Fig. 13b, and subsequently provided to the transform generator – col. 2, lines 54-57. Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a partial key (i.e. Fig. 13B – each key is split into one partial key which is comprises a majority (i.e. 128 of 196) of consecutive bits). Each partial key helps to form the constituent elements of, and correspond to, its respective original key (196-bits). The partial keys (128-bit) require less memory than the original keys (128-bits).

Referring to Fig. 9, each key is stored as an entry in the hash table (i.e. memory table with 10(exp)8 keys) – The transform generator determines an address and a confirmer for each key (col. 2, lines 47-48. The information determined from each of the original keys (or partial key as shown in Fig. 13b) is used to prevent the occurrence of collisions (i.e. hashing conflicts) – col. 2 lines 21-30);

a hash function block coupled to the hash block that applies any polynomial to a full key and generates a partial key and a hash value which is used to point to one of the plurality of partial keys stored in the hash table wherein the plurality of partial keys include saved bits comprising a consecutive, sequential strings of bits derived from the plurality of original full keys, and wherein the hash value includes bits from the full key that are not included in any

of the partial keys (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to the partial key in Fig 13b. – the original key (element 324) is split into a partial key, and the hash function is applied. Additionally, referring to Fig. 13b, the original key (element 324) is comprised of a partial key (elements A+B). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-49). Additionally note that addressing and pointer information is stored directly in the memory as per col. 2, line 66 through col. 3, line 1. Also note as per col. 2, lines 54-65, that the transform generator uses polynomial code to generate address and confirmer information for the key. The full key is transformed (i.e. the partial key plus the last 64 bits as depicted in Fig. 13B as element 324). Since the bits of element 324 are NOT part of the partial key, Brandin teaches a hash value as including bits from the full key that are not included in any of the partial keys as recited by Applicant in the instant claims. Additionally note, the claim requires the hash value to equal "a number" of bits of the original key minus "a number" of bits of the partial key. Note "a number" is broadly construed as "any number" rather than "total number" of bits. Based on Examiner's broadest reasonable interpretation pursuant to MPEP § 2111, the number of Brandin's hash value equals a number of bits in the original key minus a number of bits in the partial key (e.g. all 196 bits of the original key minus 0 bits of the partial key = 196 which is the size of the hash value); and

a processor that identifies keys and conflicts by comparing one of the plurality of partial keys to the partial key comprising a majority of bits of the full key (again, the partial key comprises a majority of consecutive bits of the original key (element 324, A+B), Brandin's system inherently contains a processor (whether it be either hardware, software, or a combination of the two) to perform the comparison.

Despite these teachings Brandin fails to teach NOT storing the hash value in the hash table.

Biran however teaches managing the pointer and hash value functionality in the I/O adapter (Fig. 3, element 38 – translation table), rather than in the system memory (i.e. not in the hash table itself).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

As for claim 28, Brandin teaches the multiple partial keys correspond to an equal number of multiple input keys (each original key (element 324, A+B+C) contains one

partial key (element 324 (A+B). The partial key is 128-bits and the original key is 196-bits).

As for claim 29, Brandin teaches multiple partial keys as each being selectable according to a different hash value derived from one of the equal number of multiple input keys (Fig. 1, the keys stored in memory are uniquely selected based on the hash value).

As for claim 30, Brandin teaches comparing of the partial key as comprising reading less data than that contained in the original key (each partial key contains less bits than the original key so less bit are read for the comparison).

As for claim 2, Brandin teaches the partial key from the hash table corresponding to the hash value includes saved bits comprising a consecutive, sequential string of bits, that is a subset of the original key where the subset includes a majority of bits of the original key (referring to Fig. 13b, the original key (element 324, A+B+C) contains one partial key (elements 324, A+B). The bits in the partial key are stored in a sequential line (based on the key length), each containing more than half of bits than the original key – col. 7, lines 14-49).

As for claim 3, Brandin teaches the stored partial key comprises a number of bits equal to or more than a number of bits of the original key minus a number of bits of the hash value (referring again to Fig. 13B, partial key A+B (element 3324) is input into the LFSR to generate a hash value (transform) which is equal or greater in size to the partial key. Since the partial key is more than half the original key's size, the partial key is equal to or great than the size of the original key minus the hash value – col. 7, lines

14-49). Again, "a number" is broadly construed as "any number" rather than "total number" of bits. Based on Examiner's broadest reasonable interpretation pursuant to MPEP § 2111, the number of Brandin's hash value equals a number of bits in the original key minus a number of bits in the partial key.

As for claim 4, Brandin teaches the hash function as being is implemented by a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

As for claim 8, Brandin teaches the apparatus of Claim 7, wherein the hash table comprises a 2(exp)N hash table size (referring to Fig. 3, the store table example used (element 50) contains 16 entries (i.e. N=4)).

As for claim 9, Brandin teaches the apparatus of Claim 7, wherein the one of the plurality of partial keys stored in the hash table comprises a number of bits equal to or more than the number of bits of the full key minus a number of bits of the hash value (referring again to Fig. 13B, partial key (element 324, A+B) is input into the LFSR to generate a hash value (transform) which is equal in size to the partial key (X(A) + X(AB)). Since the partial key is more than half the original key's size, the partial key is equal to or greater than the size of the original key minus the hash value – col. 7, lines 14-49). Again, "a number" is broadly construed as "any number" rather than "total number" of bits. Based on Examiner's broadest reasonable interpretation pursuant to MPEP § 2111, the number of Brandin's hash value equals a number of bits in the original key minus a number of bits in the partial key

As for claim 10, Brandin teaches the apparatus of Claim 7, wherein the hash function block comprises a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

As for claim 35, Brandin teaches the stored partial keys include a majority of data bits of the corresponding input keys (the partial key, element 324 (A+B) is a majority subset of the full key (A+B+C), element 324).

As for claim 36, Brandin teaches partial key includes multiple consecutive sequential strings of bits derived from the input key, separated by the bits of the hash value (referring to Fig. 3, the table stores the key and the transforms. The partial keys comprise consecutive bits which are separately stored from the hash value.

As for claim 37, Brandin teaches the hash value includes an address location associated with the input key (the memory store as depicted in Fig. 1 stores the pointer information to locate the key).

As for claim 38, Brandin teaches wherein the hash value includes bits from the input key that are not included in any of the stored partial keys (col. 2, lines 54-65 that the transform generator uses polynomial code to generate address and confirmer information for the key. The full key is transformed (i.e. the partial key plus the last 64 bits as depicted in Fig. 13B as element 324). Since the bits of element 324 are NOT part of the partial key, Brandin teaches a hash value as including bits from the full key that are not included in any of the partial keys as recited by Applicant in the instant claims).

5. Claims 6 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin (US Patent 6,493,813) and Biran (US Patent 6,345,347) as applied to claims 1 and 7 above respectively, and in further view of Ji (US PG Publication 2005/0086363 A1).

As for claim 6, Brandin in further view of Biran teaches:

> reading a result from the hash table corresponding to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49, and the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11).

Brandin in further view of Biran however fails to teach forwarding a packet of data according to the result read from the memory.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

As for claim 14, the combined teachings of Brandin and Biran fail to teach the apparatus of claim 7 further comprising a forwarding engine coupled to the hash table, wherein the forwarding engine forwards a data packet according to information read from the hash table at an address corresponding to the one of the plurality of partial keys stored in the hash table.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for the combined teachings of Brandin and Biran to further implement Ji's traffic flow management system in order for them to send information referenced by his memory store, as a series of packets. By doing so, they would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

6. Claims 5, 13 and 31-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin (US Patent 6,493,813) and Biran (US Patent 6,345,347) as applied to claims 1, 7 and 27 above, and in further view of Bryg et al., hereinafter Bryg (US Patent 6,430,670 B1).

As for claim 5, the combined teachings of Brandin and Biran fail to teach applying a reverse function on the partial key from the memory corresponding to the hash value to generate the original key.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21.

As for claim 13, though the combined teachings of Brandin and Biran fail to teach the apparatus of claim 7 further including a reverse function generator coupled to the hash table wherein the reverse function generator restores the full key based on the one of the plurality of partial keys stored in the hash table and the hash value.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed, by applying the reverse hash function on the hash result and the hash identifiers for key restoration – col. 8, lines 4-21. Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

As for claims 32 and 33, though Brandin and Biran teach all of the limitations of claim 27, they fail to teach recovering the original by combining the key with a hash value via a reverse function generator coupled to the memory wherein the reverse function generator restores the full key based on the one of the plurality of partial keys stored in the memory and the hash value.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed, by applying the reverse hash function on the hash result and the hash identifiers for key restoration – col. 8,

lines 4-21. Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

It would have been obvious to one of ordinary skill in the art at the time of the invention for the combined teachings of Brandin and Biran to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, they would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide them with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn, they would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

As for claim 31, Brandin teaches the hash value corresponds to a single entry in the hash table (Fig. 1, each address contains pointer and confirmer information unique to the keys).

7. Claims 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin (US Patent 6,493,813) and Biran (US Patent 6,345,347) as applied to claim 10 above, and in further view of Rajski et al., hereinafter Rajski (US PG Publication 2002/0016806 A1).

As for claims 11 and 12, the combined teachings of Brandin and Biran fail to teach the LFSR as corresponding to either a Fibonacci, or a Galois version.

Rajski however teaches a method for synthesizing linear finite state machines, which includes both the Fibonacci, or a Galois versions – paragraph 0002, lines 17-20 and paragraph 0003, lines 1-4 – both types are described in his teachings.

It would have been obvious to one of ordinary skill in the art at the time of the invention for the combined teachings of Brandin and Biran to further implement Rajski's method for synthesizing linear finite state machines for his own LFSRs. By doing so, they would be able to more efficiently implement his LFSR with fewer levels of logic, and a lower internal fan-out of the circuitry, as taught by Rajski (paragraph 0012, lines 1-18).

Response to Arguments

8. Applicant's arguments with respect to rejections previously set forth under 35 U.S.C. §§ 102 and 103 have been fully considered but they are not persuasive.

9. Response to arguments under the heading *35 USC §102 Rejection of claims 7-10, 34, 35, 37 and 38*:

With respect to claim 7, Applicant asserts, "Examiner's characterization of the key 324 as being a partial key is incongruous with Brandin's describing the key 324 as including 196 bits that are divided into three equal 64 bit portions. Applicant similarly submits that it is improper to characterize combined multiple transforms or portions (e.g. A and B) of the key 324 as being a partial key. Specifically, Applicant respectfully disagrees with the Examiner's interpretation of transform (AB) 328 as including 128 bits. Brandin clearly identifies that the three transforms of 64 bits each as being

concatenated to form the concatenated transform 332. That the Examiner's interpretation is further flawed is evidenced by the fact that the transform (ABC) 330 is not described as being the concatenated transform (of three 64 bit portions) but rather is itself combined with the transforms 326 and 328 to make the concatenated transform 332 (col. 7, lines 40-45 and FIG. 3B)."

This argument however is not persuasive. The claim requires, *inter alia*, "a plurality of partial keys used to determine hashing conflicts, wherein the plurality of keys correspond to a plurality of original full keys". As discussed in the previous Office action, Brandin clearly discloses a full key (representative of one of many full keys taught by Brandin (col. 2, lines 54-57) consisting of 196 bits in Fig. 13B (portions A, B and C)). Each original key consists of one partial key (the first 128 bits of the full key labeled A+B by Brandin). It appears that Applicant disagrees with Examiner's labeling of Brandin's partial key, however the fact remains that this partial key (A+B portions in Fig. 13B) are "used to determine hashing conflicts" as recited by Applicant in the instant claim, therefore it may be broadly construed as a "partial key" based on Examiner's "broadest reasonable interpretation consistent with Applicant's specification" pursuant to MPEP § 2111.

Applicant continues, "[f]urthermore, transforms are not partial keys. Neither the three portions nor the three transforms describe a partial key that comprises a majority of bits of a full key, as recited by claim 7. Rather, any of the portions or transforms including 64 bits each are approximately one third of the 196 bit key 324 or the extended transform 332. One third is not a majority."

Again this argument is not persuasive, as Examiner mapped two thirds of the full key (not one third as suggested by Applicant) of the full key to read on the partial key. Applicant asserts that one third is not a majority, and Examiner agrees. However this argument is moot since Examiner mapped two thirds of the full key to Applicant's partial key.

Applicant lastly asserts, "[a]mended claim 7 recites that the hash value includes bits from the full key that are not included in any of the partial keys. Brandin, on the other hand, describes that all the bits of the key 324 are included in the transforms 326, 328, 330 (col. 7, lines 40-45). On page 15, the Examiner identifies the transform as a hash value. It does not logically follow that the hash value can include bits from the full key that are not included in any of the partial keys, if the transforms include all the bits of the full key and also function as a hash value. As Brandin fails to disclose or teach all the features recited in amended claim 7, claim 7 is believed to be allowable over Brandin."

This argument however is not persuasive. Examiner maintains as per col. 2, lines 54-65 that the transform generator uses polynomial code to generate address and confirmer information for the key. The full key is transformed (i.e. the partial key plus the last 64 bits as depicted in Fig. 13B as element 324). Since the bits of element 324 are NOT part of the partial key (e.g. portion C), Brandin teaches a hash value as including bits from the full key that are not included in any of the partial keys as recited by Applicant in the instant claims.

10. Response to arguments under the heading *35 USC §103 Rejection:*

With respect to claim 1, Applicant asserts, "Examiner identifies a hash value as being an address 28 and confirmer 30 as applied to FIG. 13B. Applicant respectfully submits that the address 28 and confirmer 30 do not include a number of bits equal to a number of bits of the original key minus a number of bits of the partial key. Rather, the address 28 and confirmer 30 are provided as part of the transforms (col. 2, lines 31-36). As the Examiner has also identified the transform as a hash value (page 15 of the Office Action), Applicant points out that this also would fail to disclose wherein the hash value is not stored in a hash table, as recited by amended claim 1. All the data for the transforms of Brandin are saved at their respective addresses in memory store 200 (see FIGS. 9 and 10)."

This argument however is not persuasive. The claim requires the hash value to equal "a number" of bits of the original key minus "a number" of bits of the partial key. Note "a number" is broadly construed as "any number" rather than "total number" of bits. Based on Examiner's broadest reasonable interpretation pursuant to MPEP § 2111, the number of Brandin's hash value equals a number of bits in the original key minus a number of bits in the partial key (e.g. all 196 bits of the original key minus 0 bits of the partial key = 196 which is the size of the hash value).

Applicant further asserts, "[a]s acknowledged by the Examiner at page 13, Branding (sic) fails to disclose comparing keys or where the hash value is not stored in memory. Instead the Examiner cites Biran to disclose these features. Biran directly compares the entire keys, however neither key is a partial key as recited by claim 1. Even assuming Biran teaches hash values that are not saved in a hash table, Biran fails

to cure the deficiencies of Brandin since the hash values of Biran do not include a number of bits equal to a number of bits of the original key minus a number of bits of the partial key."

This argument however is not persuasive as Examiner maintains the combination of Brandin in further view of Biran teach all limitations of claim 1. Applicant is reminded pursuant to MPEP § 2145 (IV.), "one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981) ." Examiner maintains that Brandin in further view of Biran discloses comparing partial keys and NOT storing the hash value in the hash table. Applicant's choice to attack these references individually is hence not persuasive, as it does not fully address Examiner's properly asserted *prime facia* case of obviousness.

Lastly, Applicant assets, "[t]here is no teaching or suggestion from this combination that a hash value could be both not stored in a hash table while also include a number of bits of the original key minus a number of bits of the partial key", and that "[o]ne skilled in the art would not be able to determine how Brandin's transforms could be combined with Biran's hash values to teach the hash values recited by claim 1" Continuing, Applicant asserts, "combining the references to teach the features of claim 1 would make the transforms of Brandin inoperable. If bits are removed from Brandin's transforms to create a hash value, then the Brandin transforms would no longer operate to resolve a hashing conflict in the event that some or all of the remaining bits between partial keys matched. Brandin instead teaches that the

transform generator must provide a unique transform for every possible key of a certain length (col. 2, lines 40-42). Brandin teaches that the extended transform 332 must therefore include all the bits of the key."

These arguments however are not persuasive. Examiner maintains pursuant to MPEP § 2145 (III.), "[t]he test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference.... Rather, the test is what the combined teachings of those references would have suggested to those of ordinary skill in the art." *In re Keller*, 642 F.2d 413, 425, 208 USPQ 871, 881 (CCPA 1981)." As per the rejection of claim 1, Brandin in fact renders a hash value as including a number of bits equal to a number of bits of the original key minus a number of the bits of the partial key obvious as per the arguments and rejections *supra*. Further, though Examiner concedes that Brandin fails to teach NOT storing his hash value in the hash table, Examiner maintains that Brandin in further view of Biran teach this limitation as per the rejection above, that that it would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further include Biran's address protection system using a hardware-defined application key to improve the chances of avoiding hashing conflicts as per Biran's teachings in col. 2, lines 58-67.

Conclusion

11. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

12. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Craig E. Walter whose telephone number is (571) 272-8154. The examiner can normally be reached on 8:30a - 5:00p M-F.

14. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung S. Sough can be reached on (571) 272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

15. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Craig E Walter
Examiner
Art Unit 2188

CEW

HYUNG S. SOUGH
SUPERVISORY PATENT EXAMINER
11/02/07